

AF MW

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:

Allavarpu, et al.

Serial No. 09/552,984

Filed: April 21, 2000

For: System and Method for  
Event Subscriptions for  
CORBA Gateway

§ Group Art Unit: 2151  
§  
§ Examiner: Dinh, Khanh Q.  
§  
§ Atty. Dkt. No.: 5181-48200  
§ P4498

CERTIFICATE OF MAILING  
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert  
Name of Registered Representative

July 13, 2005  
Date

[Signature]  
Signature

**APPEAL BRIEF**

**Mail Stop Appeal Brief - Patents**  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed May 13, 2005, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

## **I. REAL PARTY IN INTEREST**

As evidenced by the assignment recorded at Reel/Frame 010993/0884, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

## **II. RELATED APPEALS AND INTERFERENCES**

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

## **III. STATUS OF CLAIMS**

Claims 1-9, 11-22, 24-35, and 37-39 stand finally rejected. Claims 10, 23 and 36 are objected to as depending from rejected independent claims, but would otherwise be allowable if rewritten in independent form. The rejection of claims 1-9, 11-22, 24-35, and 37-39 is being appealed. A copy of claims 1-9, 11-22, 24-35, and 37-39 as currently pending is included in the Claims Appendix herein below.

## **IV. STATUS OF AMENDMENTS**

An amendment to claims 6, 19 and 32 was submitted on April 11, 2005 as part of the Response to Final Office Action and the Examiner indicated that the amendment would be entered (Advisory Action, April 27, 2005). The Claims Appendix included herewith therefore reflects the state of the claims including the amendments mentioned above.

An additional amendment, submitted via Facsimile on July 12, 2005, amends claims 27 – 39 to recite a tangible, computer accessible medium rather than a carrier medium. The Examiner has not indicated whether this amendment will be entered. The

Claims Appendix included herewith reflects the state of the claims prior to this amendment.

## **V. SUMMARY OF CLAIMED SUBJECT MATTER**

Independent claim 1 is directed to a network management system including an event gateway coupled to one or more managed objects and which is configured to deliver events generated by the managed objects to one or more managers. A manager may send requests to a managed object and the managed object may send events to the manager. A managed object may initiate or generate an event, such as a notification, warning, or alarm concerning a status or occurrence related to the managed object. For example, a managed object may generate an event when a monitored device experiences an error that causes it to suspend operation. In another example, a managed object may generate an event if usage of a particular resource reaches a particular threshold, such as a printer running low on toner. An event gateway may be coupled between the manager and the managed object and deliver both the requests and events. In one embodiment, an event gateway may be CORBA based and may deliver requests and events between CORBA-based manager and managed objects. In some embodiments, an event gateway may be part of a larger gateway. For example, a CORBA-based event gateway may be a component of a general CORBA gateway. (See *e.g.*, FIG 2, 3, 4, 7, 8 and 15; page 13, line 27 – page 14, line 20; page 19, line 21 – page 20, line 8; page 21, lines 7 – 15; page 35, line 12 – page 26, line 11).

The network management system of claim 1 also includes a platform-independent interface to the event gateway. For example, the use of IDL may provide a platform-independent interface to deliver events generated by managed objects to manager applications through an event gateway. The event gateway may be configurable to communicate with the managers through the platform-independent interface to deliver the events generated by the managed objects. For example, in one embodiment, the gateway may be CORBA based and may utilize various protocols and formats, such as Internet Inter-Object Protocol (IIOP) and Interface Definition Language (IDL) and may translate

between various formats and protocols. (See *e.g.*, FIG. 2, 3, 4, 7, 8, and 15; page 11, lines 3 – 25; page 12, lines 23- 26; page 20, lines 20 – 25; page 21, lines 7 – 24; page 22, lines 3 – 13; page 45, lines 13 – 21).

The event gateway of claim 1 also includes a plurality of event distribution server sinks configured to receive events generated by the managed objects and distribute the events to the managers such that one of the managers received events from a plurality of different ones of the event distribution server sinks. For example, the event gateway may leverage existing event distribution server sinks and may in some embodiments CORBA-enable event distribution sinks. The event gateway may deliver events from multiple sinks to a single manager application instead of having all managers receive their events from one event sink. Thus, the event gateway and the plurality of event distribution sinks may provide load balancing. The event gateway may utilize a platform-independent interface to deliver events generated by managed objects to manager applications. (See, *e.g.*, FIG. 2, 3, 4, 6, 7, 8 and 15; page 11, line 27 – page 12, line 26; page 12, line 28 – page 13, line 25; page 25, line 12 – page 26, line 11; page 28, lines 12- 24;

The event gateway is also configurable to provide the managers with subscriptions to the events as a function of event criteria specified by the managers, whereby events meeting the specified event criteria are delivered and events failing to meet the specified event criteria are filtered out. Clients, such as managers, may subscribe to events based on criteria, such as object class, object instance, and event type. For example, a print manager application may subscribe to all events relating to toner supply of all laser printers. Managers may subscribe or unsubscribe to events using IDP APIs provided by the event gateway. The event gateway and/or the server sinks may provide the capability to filter events according to criteria presented when a manager subscribes to events. For example, an event distribution server sink may determine which managers are to receive a particular event and route the event to the appropriate manager. Thus, in some embodiments, events may be filtered at the sink level rather than on the client or manager side, which may reduce network traffic. In some embodiments, the event gateway may manage event subscriptions using services provided by an event

port registry server. (*See, e.g.*, FIG. 2, 3, 4, 6, 7, 8 and 15; page 12, line 27 – page 13, line 26; page 20, lines 10 – 25; page 21, line 7 – page 22, line 13; page 22, line 16 – page 24, line 11).

Independent claim 14 is directed to a network management method including registering a subscription of a manager application to events generated by managed object by specifying event criteria to an event gateway. In general, the method recited by claim 14 may include the capability to filter events according to criteria provided by client (or manager) applications. A client may specify such event criteria, such as object class, object instance, or event type, when requesting a subscription to events. Event subscriptions may be managed via services provided by an event port registry server. The event gateway is configurable to communicate with the manager application through a platform-independent interface, such as via IIOP or IDL, provided by an event gateway between the manager application and managed objects. As noted above, an event gateway may leverage existing event distribution server sinks and may in some embodiments CORBA-enable event distribution sinks. For more information regarding event subscriptions and filter and about a platform-independent interface provided by an event gateway, please the discussion of claim 1 above. (*See, e.g.*, FIG. 2, 3, 4, 6, 7, 8 and 15; page 12, line 27 – page 13, line 26; page 20, lines 10 – 25; page 21, line 7 – page 22, line 13; page 22, line 16 – page 24, line 11).

The method of claim 14 also includes generating events including events matching the specified event criteria, determining whether the specified event criteria are met for each of the generated events and delivering each event for which the specified event criteria are met. The event gateway and/or the server sinks may provide the capability to filter events according to criteria presented when a manager subscribes to events. For example, an event distribution server sink may determine which managers are to receive a particular event and route the event to the appropriate manager. Thus, in some embodiments, events may be filtered at the sink level rather than on the client or manager side, which may reduce network traffic. The events are delivered from a plurality of different event distribution server sinks of the event gateway to the manager

application. For instance, an event gateway may deliver events from multiple sinks to a single manager application instead of having all managers receive their events from one event sink, thus providing load-balancing for event delivery. (*See, e.g.*, FIG. 2, 3, 4, 6, 7, 8 and 15; page 12, line 27 – page 13, line 26; page 20, lines 10 – 25; page 21, line 7 – page 22, line 13; page 22, line 16 – page 24, line 11).

Independent claim 27 is directed to a medium including program instructions that are computer-executable to perform the method recited by claim 14, described above. For a detailed discussion of the method performed by the program instructions of claim 27, please refer to the discussion of claim 14 above.

## **VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 1-9, 11-22, 24-35 and 37-39 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Barker et al. (U.S. Patent 6,363,421) (hereinafter “Barker”) in view of Sampat et al. (U.S. Patent 6,279,029) (hereinafter “Sampat”).

## **VII. ARGUMENT**

### **Claims 1 – 6 and 13:**

Regarding claim 1, contrary to the Examiner’s assertion, Barker in view of Sampat does not teach or suggest an event gateway that comprises a plurality of event distribution server sinks configured to receive events generated by managed objects and distribute the events to one or more managers such that one of the managers receives events from a plurality of different ones of the event distribution server sinks. Barker discloses a method for remotely managing a plurality of network elements of a telecommunications network through a special communications link. A management computer is connected to an element management system server 32 through a communication link including the Internet. At least one of the network elements 14 is also coupled to the element management server 32 through the internet and is managed via communications conveyed through the element management server 32 between the

management computer and the at least one network element 14. Sampat teaches a client/server system that sends and receives multicast media data streams over a network.

The Examiner admits that Barker “does not disclose a plurality of server sinks configured to receive events generated by the managed objects and distribute the events to one or more managers” and asserts that Sampat teaches such a plurality of sinks. Appellants respectfully disagree with the Examiner’s characterization of Sampat. However, Sampat does not teach or suggest event distribution server sinks configured to *receive events generated by managed objects and distribute the events to the one or more managers* such that one of the managers receives events from a plurality of different one of the event distribution server sinks. Instead, Sampat teaches a client/server architecture for network based multicast systems. Sampat’s client/server architecture includes a media services manager and media services that send and receive multicast media data streams over a network (Sampat, Abstract, column 2, lines 25-36).

Sampat uses source and sink media service providers (MSPs) that assist in the receipt of and playing of data streams, respectively (Sampat, column 9, line 24). **However, Sampat’s sink MSPs are not event distribution server sinks that receive and distribute events.** In contrast, Sampat teaches that sink MSPs are used to transfer received media data (text, video, and audio, for example) to appropriate hardware drivers for local playing. (Sampat, Fig. 16, items 1614, 1618, and 1622, Fig. 18, items 1814, 1414, and 1822). Specifically, Sampat teaches that media server provider sinks are used to play multicast data either received from the network or locally recorded (Sampat, column 14, lines 18-28). Additionally, Sampat describes how “[v]ideo sink MSP 1814 and text sink MSP 1822 receive a video data stream and a text data stream, respectively, from MSM 1808 and transmits the video and text data to display driver ... for display on [a] monitor” (Sampat, column 14, lines 38 – 41). Similarly, Sampat teaches that an audio sink transmits received audio data to an audio driver for playing on audio hardware (Sampat, column 14, lines 42-44). Thus, Sampat teaches the use of media service provider sinks to transfer received media data to appropriate hardware device drivers. The source and sink media service providers of Sampat have nothing to do with receiving

and distributing events of any kind, let alone events generated *by managed objects* and *distributed to one or more managers*. The Examiner cites column 13, line 62 to column 14, line 57 of Sampat for support. However, as shown above, this passage does not teach event distribution sinks, but instead teaches media service provider sinks that transfer media data to appropriate hardware device drivers.

Additionally, sink MSPs, as taught by Sampat, each receive data from a *single* data stream and forward it to a *single* device driver. Sampat teaches that media service manager (MSM) starts a sink MSP for each data stream (Sampat, column 14, lines 6-11). Further, as shown above, each sink MSP transmits data to a specific device driver (Sampat, column 14, lines 38-44). Thus, Sampat's sink MSPs do not distribute the data that they receive, but merely transfer it from a specific input to a specific output. Sampat teaches that a single process, the media services manager (MSM), actually determines how to route incoming data. For example, Sampat states, "[u]pon receiving new data from the network, the MSM transmits the data to the appropriate MSP." (Sampat, column 15, lines 51-53). Thus, Sampat would in no way suggest modifying Barker to distribute events to one or more managers such that one of the managers receives events from a plurality of different ones of the event distribution server sinks.

In the Response to Arguments section, the Examiner responds to the above remarks regarding Sampat's failure to teach a plurality of event distribution server sinks. Specifically, the Examiner refers to Sampat's Media Service Providers MSP and cites figure 18 and column 9, line 10 - column 10, line 46. The Examiner also argues in the Advisory Action that Sampat teaches MSPs to provide monitoring services capabilities to server applications, further citing column 13, line 62 – column 14, line 57. However, as noted above, Sampat teaches the use of media service provider sinks to transfer received media data to appropriate hardware device drivers. Additionally, Sampat's teachings regarding providing monitoring services does not involve event distribution server sinks configured to *receive events generated by managed objects* and *distribute the events to the one or more managers*. Instead, Sampat's monitoring services consists of the MSM sending a copy of the media data received by a sink to another sink MSP (Sampat,



column 12, lines 39 – 41). Furthermore, Sampat teaches that the media data may flow from a source MSP to a sink MSP for monitoring the processing of data received from an external source (Sampat, column 8, line 61 – column 9, line 3). Thus, Sampat's monitoring does not include receiving or sending events, as suggested by the Examiner. Nowhere does Sampat teach or suggest that his monitoring functionality includes receiving events generated by managed objects and distributing those events to managers. Instead Sampat teaches that a copy of the media data stream may be sent to another MSP sink for monitoring purposes.

**The source and sink media service providers of Sampat clearly have nothing to do with receiving and distributing events of any kind, let alone events generated by managed objects and distributed to one or more managers.**

Appellants also submit that the combination of Barker and Sampat suggested by the Examiner would not result in an event gateway that comprises a plurality of event distribution sinks configured to receive events generated by the managed objects and distribute the events to the one or more managers. Instead any modification based on Sampat of Barker's system for managing network elements to use the sink media service providers of Sampat would at most result in a system for managing media service providers. However, since neither Barker nor Sampat teach the use event distribution sinks (that receive events from managed objects and distribute the events to one or more managers), no combination of Barker and Sampat would suggest such event distribution sinks.

In the Response to Argument section of the Final Office Action, the Examiner argues, "Appellant simply asserts 'the references would not result in the claimed invention'." The Examiner further states, "[t]his quote is the extent of explanation provide by Appellant in support of claims 1, 14, and 27." Contrary to the Examiner's assertion, Appellants have clearly explained why the combination of Barker and Sampat would not result in the claimed invention. The Examiner is completely ignoring the majority of Appellants' arguments regarding claims, 1, 14 and 27 submitted in the

Response to Office Action filed August 20, 2004. For instance, Appellants argued previously, and currently above, that Barker in view of Sampat does not teach or suggest an event gateway comprising a plurality of event distribution server sinks configured to receive events generated by the managed objects and distribute the events to the one or more managers such that one of the managers receives events from a plurality of different ones of the event distribution server sinks. The Examiner has even responded to this particular argument in the Response to Arguments section, as noted above. Thus, Appellants find no basis for the Examiner's assertion that Appellants' response "is insufficient to satisfy the requirement of specific argument to have the claims considered for patentability."

In further response to Appellants' argument above, the Examiner argues, in the Advisory Action, that the Media Service Providers (MSPs) of Sampat correspond to event distribution server sinks and that the Sampat's Media Service Managers (MSMs) correspond to managers. Please see the discussion above for a detailed discussion regarding why Sampat's MSPs cannot be considered event distribution server sinks. Regarding Sampat's Media Service Managers, nowhere does Sampat describe his MSPs as delivering events generated by managed objects to the MSM. Instead, as illustrated in FIG. 18, the MSM delivers media data to the MSPs. The Examiner has not cited any portion of Sampat that describes events being delivered by the MSPs to an MSM.

Furthermore, Barker clearly teaches a single event distributor 140 (Barker -- Fig. 4). Barker teaches that this single event distributor "is responsible for filtering and routing of all events in the system" (emphasis added, Barker, column 17, lines 5-6). Thus, **Barker teaches away** from a plurality of event distribution server sinks configured such that one of the managers receives events from a plurality of different ones of the event distribution server sinks.

Additionally, Appellants strongly disagree with the Examiner's assertion that Barker's applications 44 and 50 are managed objects that generate events. **Barker clearly describes applications 44 and 50 as client applications that are distinct from**

**the managed objects** (See, e.g., Barker, column 11, lines 47-60; column 13, line 46 – column 16, line 12; column 39, line 55 – column 40, line 16). At the top of Fig. 6, Barker defines an example of a managed object as a network element. Thus, in Fig. 2, the managed objects are represented by network element 14, *not client applications 44 and 50*. Anyone of ordinary skill in the art reading Barker would understand that client applications 44 and 50 are not managed objects in Barker's system. **Furthermore, there is no teaching in Barker that client applications 44 and 50 generate events.** The Examiner's cited passages include absolutely no mention at all that client applications 44 and 50 generate events received by element management system server 32. Appellants request that the Examiner quote the exact column and line numbers where he believes Barker teaches that client applications 44 and 50 are managed network elements and generate the events described in Barker. **Although Appellants have repeatedly asserted this argument, Appellants note that the Examiner has never supplied any rebuttal of this specific argument.** Instead, in the Advisory Action, the Examiner argues that Barker teaches that all resources and elements that can be managed are represented in the system as a managed object. This argument by the Examiner merely supports Appellants' argument. Since Barker does not represent applications 44 and 50 as managed objects, applications 44 and 50 are clearly not managed objects that generate events, contrary to the Examiner's assertion.

#### **Claim 7:**

In regard to claim 7, contrary to the Examiner's assertion, Barker in view of Sampat fails to teach that the managed objects comprise one or more objects **corresponding to a telephone network**. In contrast, Barker discloses a system client that is connected to a network element and element management system client *through* a public switched telephone network (Barker, column 3, lines 48-53). Additionally, Barker teaches the use of a telephone system network through the computer Internet and a telephonic link for a system client to connect to the system server (Barker, column 3, lines 54-62). The Examiner cites column 3, line 47 – column 4, line 36; and column 7, line 38 – column 8, line 67 of Barker. However, neither of the Examiner's cited passages

describes any of Barker's managed objects (network elements) as corresponding to a telephone network. Instead, the first cited passage generally describes the hardware and environments on which Barker's system may be implemented. The second of the Examiner's cited passages describes the various components of Barker's element management system, but does not mention anything about a managed object corresponding to a telephone network.

Furthermore, Barker's use of the phrase "network elements of a telecommunication network" (See, Barker, Title, and brief descriptions of FIG 1A, 1B, and 1C, column 2, lines 50-65) does not imply that one or more of Barker's network elements correspond to a telephone network. Barker is clearly referring to network element *residing on* a telecommunications network. For instance, when discussing FIG. 1B, Barker describes his system as a "method for managing the network element 14 *in a telephonic network*" and continues, "[n]etwork element 14 is connected *through* a telephonic computer network 35 to a computer internet 36" (emphasis added, Barker, column 3, lines 53-58). In other words, network element 14 is not illustrated as *corresponding to* a telephone network, but rather network element 14 is illustrated as coupled to and communicating over a telephone network. Hence, Barker discloses using a telephonic connection between clients and servers but fails to disclose anything regarding managed objects comprising one or more objects corresponding to a telephone network. This is made clear when figures 1A, 1B and 1C are viewed together. Barker is illustrating that fact that his system may be implemented (e.g. his element management server may communicate with network elements) over various types of communication networks, such as public switched telephone network 33 (Figure 1A), telephonic system network 37 (Figure 1B), and a local area network (Figure 1C).

None of the managed objects in Barker *correspond to a telephone network* themselves, but instead communicate using a telephone network. Sampat is not relied by the Examiner in the rejection of claim 7, nor does Sampat overcome any deficiency of Barker regarding managed objects corresponding to a telephone network. Thus, Barker

in view of Sampat clearly fails to teach wherein the managed objects comprise one or more objects corresponding to a telephone network.

**Although Appellants have repeatedly asserted this argument, Appellants note that the Examiner has never supplied any rebuttal to this specific argument.**

**Claim 8:**

Regarding claim 8, contrary to the Examiner's assertion, Barker in view of Sampat fails to teach or suggest wherein the managed objects comprise an object corresponding to a telecommunications device. The Examiner cites column 3, line 47 – column 4, line 36; and column 7, line 38 – column 8, line 67 of Barker. However, neither of the Examiner's cited passages describes any of Barker's managed objects (network elements) as corresponding to a telecommunications device. Instead, the first cited passage generally describes the hardware and environments on which Barker's system may be implemented. The second of the Examiner's cited passages describes the various components of Barker's element management system, but does not mention anything about a managed object corresponding to a telecommunications device.

Furthermore, as noted above regarding claim 7, Barker's use of the phrase "network elements of a telecommunication network" (See, Barker, Title, and brief descriptions of FIG 1A, 1B, and 1C, column 2, lines 50-65) does not imply that one or more of Barker's network elements correspond to a telecommunications device. Appellants submit, that Barker is referring to network element residing on a telecommunications network. For instance, when discussing FIG. 1B, Barker describes his system as a "method for managing the network element 14 *in a telephonic network*" and continues, "[n]etwork element 14 is connected *through* a telephonic computer network 35 to a computer internet 36" (emphasis added, Barker, column 3, lines 53-58). In other words, network element 14 is not illustrated as corresponding to a telecommunications device, but rather network element 14 is illustrated as coupled to and communicating over a telephone network. Hence, Barker discloses using a telephonic

connection between clients and servers but fails to disclose anything regarding managed objects comprising one or more objects corresponding to a telecommunications device. This is made clear when figures 1A, 1B and 1C are viewed together. Barker is illustrating that fact that his system may be implemented (e.g. his element management server may communicate with network elements) over various types of communication networks, such as public switched telephone network 33 (Figure 1A), telephonic system network 37 (Figure 1B), and a local area network (Figure 1C).

None of the managed objects in Barker *correspond to a telecommunications device* themselves, but instead communicate using a telephone network. Sampat is not relied by the Examiner in the rejection of claim 7, nor does Sampat overcome any deficiency of Barker regarding managed objects corresponding to a telecommunications device. Thus, Barker in view of Sampat clearly fails to teach wherein the managed objects comprise one or more objects corresponding to a telecommunications device.

**Although Appellants have repeatedly asserted this argument, Appellants note that the Examiner has never supplied any rebuttal to this specific argument.**

**Claim 9:**

Regarding claim 9, Barker in view of Sampat fail to teach or suggest wherein the event distribution server comprises the plurality of event distribution server sinks. The Examiner admits that Barker does not teach a plurality event distribution server sinks, but cites figure 18, column 9, line 10 to column 10, line 46 and column 13, line 62 to column 14, line 57 of Sampat. However, none of the cited passages teaches or suggests an event distribution server comprising a plurality of event distribution server sinks. As discussed above regarding claim 1, Sampat teaches media service provider sinks that transfer media data to appropriate hardware device drivers and fails to teach a plurality of event distribution server sinks. For a more detailed discussion regarding Sampat's media service provider sinks and Sampat's failure to teach or suggest event distribution server sinks, please see the remarks above regarding claim 1.

**Claim 11:**

Regarding claim 11, contrary to the Examiner's contention, Barker in view of Sampat fails to teach or suggest wherein the plurality of event distribution server sinks are operable to dispatch the events to one or more managers as a function of the subscription. The Examiner cites column 9, line 22 to column 10, line 49 and column 19, line 13 to column 20, line 59 of Barker. However, not only do neither of the cited passages mention anything regarding a plurality of event distribution server sinks operable to dispatch events to one or more managers as a function of subscriptions, the Examiner has already admitted, in the rejections of claims 1 and 9, that Barker fails to teach a plurality of event distribution server sinks (see, e.g. Final Office Action dated February 14, 2005, page 4, line 1 and page 6, lines 1-2).

Furthermore, as discussed above, regarding the rejections of claims 1 and 9, Sampat fails to teach or suggest *event distribution* server sinks. Instead, Sampat teaches media service provider sinks that transfer media data to appropriate hardware device drivers. For a more detailed discussion regarding Sampat's media service provider sinks and Sampat's failure to teach or suggest event distribution server sinks, please refer to the above discussions regarding claims 1 and 9.

**Although Appellants have repeatedly asserted this argument, Appellants note that the Examiner has never supplied any rebuttal to this specific argument.**

**Claim 12:**

Regarding claim 12, Barker in view of Sampat does not teach or suggest a plurality of event distribution server sinks that are distributed to provide load balancing of the events to the one or more managers. The Examiner cites col. 29, line 27 – col. 30, line 42; and col. 37, line 4 – col. 38, line 63, of Barker in regard to claim 12. However, these sections of Barker reveal absolutely nothing at all that corresponds to a plurality of event distribution server sinks that are distributed to provide load balancing of the events

to the one or more managers. In fact, the first cited section (Barker, col. 29, line 27 – col. 30, line 42) describes how Barker's system provides overload control by limiting the amount of network traffic through only allowing a limited number of client sessions and running applications. This cited section also describes Barker's preferred method of software version management. Neither of these topics discusses load balancing. The Examiner's second cited passage (col. 37, line 4 – col. 38, line 63) describes the major functions of Barker's event handler and goes on to provide overviews of Barker's Network Element Status Table, Network Element Status API, Event Configuration File, Admin Log, and EMAPI, none of which mention (or have anything to do with) a plurality of event distribution server sinks that are distributing to provide load balancing of the events to the one or more managers.

In the response to arguments section of the Final Office Action, the Examiner responds to the above argument by merely restating the assertion and citations from the rejection of claim 12 without providing any additional explanation or any rebuttal of Appellants' above argument. As noted above, the cited sections of Barker do not mention anything at all regarding a plurality of event distribution server sinks that are distributed to provide load balancing of the events to the one or more managers. Furthermore, the Examiner has failed in both the rejection of claim 12 and in the Response to Arguments section to point out any teaching in the cited art that equates to a plurality of event distribution server sinks that are distributed to provide load balancing of the events to the one or more managers.

**Claims 14-19, 26, 27-32 and 39:**

**Regarding claim 14, Barker in view of Sampat does not teach or suggest delivering each event for which the specified criteria are met, wherein events from which the specified event criteria are met are delivered from a plurality of different event distribution server sinks of the event gateway to the manager application.**



As described above regarding claim 1, Barker discloses a method for remotely managing a plurality of network elements of a telecommunications network through a special communications link. A management computer is connected to an element management system server 32 through a communication link including the Internet. At least one of the network elements 14 is also coupled to the element management server 32 through the internet and is managed via communications conveyed through the element management server 32 between the management computer and the at least one network element 14. Sampat teaches a client/server system that sends and receives multicast media data streams over a network.

The Examiner admits that Barker does not disclose a plurality of event distribution server sinks configured to deliver events for which the specified event criteria are met and relies upon Sampat, citing FIG. 18, column 9, line 10 – column 10, lines 46 and column 13, line 62 – column 14, line 57. However, none of the Examiner's cited passages teaches or suggests a plurality of event distribution server sinks configured to deliver events for which the specified event criteria are met to the manager application. Instead, as described above regarding claim 1, Sampat teaches a client/server architecture for network based multicast systems. Sampat's client/server architecture includes a media services manager and media services that send and receive multicast media data streams over a network (Sampat, Abstract, column 2, lines 25-36).

The Examiner argues, "Sampat discloses a plurality of server sinks configured to deliver specified event criteria generated by managed objects" (Final Office Action, page 7, lines 20-22). However, claim 14 does not recite anything about sinks delivering specifying event criteria generated by managed objects. Instead, as noted above, claim 14 refers to a plurality of different event distribution server sinks of the event gateway delivering events for which the specified event criteria are met, which is very different from delivering specified event criteria. Appellants assume the Examiner intended to argue that Sampat teaches a plurality of event distribution server sinks configured to deliver events for which the specified event criteria are met.

However, as described above, Sampat uses source and sink media service providers (MSPs) that assist in the receipt of and playing of data streams, respectively (Sampat, column 9, line 24). **Clearly, Sampat's sink MSPs are not *event distribution server sinks that receive and distribute events*.** In contrast, Sampat teaches that sink MSPs are used to transfer received media data (text, video, and audio, for example) to appropriate hardware drivers for local playing. (Sampat, Fig. 16, items 1614, 1618, and 1622, Fig. 18, items 1814, 1414, and 1822). Specifically, Sampat teaches that media server provider sinks are used to play multicast data either received from the network or locally recorded (Sampat, column 14, lines 18-28). Additionally, Sampat describes how “[v]ideo sink MSP 1814 and text sink MSP 1822 receive a video data stream and a text data stream, respectively, from MSM 1808 and transmits the video and text data to display driver ... for display on [a] monitor” (Sampat, column 14, lines 38 – 41). Similarly, Sampat teaches that an audio sink transmits received audio data to an audio driver for playing on audio hardware (Sampat, column 14, lines 42-44). Thus, Sampat teaches the use of media service provider sinks to transfer received media data to appropriate hardware device drivers. The source and sink media service providers of Sampat have nothing to do with receiving and distributing events of any kind, let alone events generated *by managed objects and distributed to one or more managers*.

Additionally, sink MSPs, as taught by Sampat, each receive data from a *single* data stream and forward it to a *single* device driver. Sampat teaches that media service manager (MSM) starts a sink MSP for each data stream (Sampat, column 14, lines 6-11). Further, as shown above, each sink MSP transmits data to a specific device driver (Sampat, column 14, lines 38-44). Thus, Sampat's sink MSPs do not *distribute* the data that they receive, but merely transfer it from a specific input to a specific output. Sampat teaches that a single process, the media services manager (MSM), actually determines how to route incoming data. **The source and sink media service providers of Sampat clearly have nothing to do with receiving and distributing events of any kind, let alone events for which the specified event criteria are met.** For a more detailed discussion regarding Sampat's MSP and how Sampat's MSP cannot be considered event distribution server sinks, please refer to the discussion of claim 1 above.

Appellants also submit that the combination of Barker and Sampat as suggested by the Examiner would not result in an event gateway that comprises a plurality of event distribution sinks configured to deliver events for which the specified event criteria are met to the manager application. Instead any modification based on Sampat of Barker's system for managing network elements to use the sink media service providers of Sampat would at most result in a system for managing media service providers. However, since neither Barker nor Sampat teach the use of event distribution server sinks (that receive deliver the events for which the specified event criteria are met to the manager application), no combination of Barker and Sampat would suggest such event distribution sinks.

In further response to Appellants' argument above, the Examiner argues, in the Advisory Action, that the Media Service Providers (MSPs) of Sampat correspond to event distribution server sinks and that the Sampat's Media Service Managers (MSMs) correspond to managers. Please see the discussion above for a detailed discussion regarding why Sampat's MSPs cannot be considered event distribution server sinks. Regarding Sampat's Media Service Managers, nowhere does Sampat describe his MSPs as delivering events generated by managed objects to the MSM. Instead, as illustrated in FIG. 18, the MSM delivers media data to the MSPs. The Examiner has not cited any portion of Sampat that describes events being delivered by the MSPs to an MSM. For a more detailed discussion regarding this argument please see the discussion of claim 1 above.

Thus, neither Barker nor Sampat, whether considered singly or in combination, teaches or suggests delivering each event for which the specified event criteria are met, wherein events for which the specified event criteria are met are delivered from a plurality of different event distribution server sinks of the event gateway to the manager application.

**Claims 20 and 33:**

In regard to claim 20, contrary to the Examiner's assertion, Barker in view of Sampat fails to teach that the managed objects comprise one or more objects corresponding to a telephone network. In contrast, Barker discloses a system client that is connected to a network element and element management system client *through* a public switched telephone network (Barker, column 3, lines 48-53). Additionally, Barker teaches the use of a telephone system network through the computer Internet and a telephonic link for a system client to connect to the system server (Barker, column 3, lines 54-62). The Examiner cites column 3, line 47 – column 4, line 36; and column 7, line 38 – column 8, line 67 of Barker. However, neither of the Examiner's cited passages describes any of Barker's managed objects (network elements) as corresponding to a telephone network. Instead, the first cited passage generally describes the hardware and environments on which Barker's system may be implemented. The second of the Examiner's cited passages describes the various components of Barker's element management system, but does not mention anything about a managed object corresponding to a telephone network.

For a more detailed discussion regarding Barker's failure to teach managed objects corresponding to a telephone network, please refer to Appellants' arguments above regarding claim 7.

None of the managed objects in Barker *correspond to a telephone network* themselves, but instead communicate using a telephone network. Sampat is not relied by the Examiner in the rejection of claim 20, nor does Sampat overcome any deficiency of Barker regarding managed objects corresponding to a telephone network. Thus, Barker in view of Sampat clearly fails to teach wherein the managed objects comprise one or more objects corresponding to a telephone network.

#### **Claims 21 and 34:**

Regarding claim 21, contrary to the Examiner's assertion, Barker in view of Sampat fails to teach or suggest wherein the managed objects comprise an object

corresponding to a telecommunications device. The Examiner cites column 3, line 47 – column 4, line 36; and column 7, line 38 – column 8, line 67 of Barker. However, neither of the Examiner's cited passages describes any of Barker's managed objects (network elements) as corresponding to a telecommunications device. Instead, the first cited passage generally describes the hardware and environments on which Barker's system may be implemented. The second of the Examiner's cited passages describes the various components of Barker's element management system, but does not mention anything about a managed object corresponding to a telecommunications device.

For a more detailed discussion regarding Barker's failure to teach managed objects corresponding to a telecommunications device, please refer to Appellants' arguments above regarding claim 8.

None of the managed objects in Barker *correspond to a telecommunications device* themselves, but instead communicate using a telephone network. Sampat is not relied by the Examiner in the rejection of claim 21, nor does Sampat overcome any deficiency of Barker regarding managed objects corresponding to a telecommunications device. Thus, Barker in view of Sampat clearly fails to teach wherein the managed objects comprise one or more objects corresponding to a telecommunications device.

#### **Claims 22 and 35:**

Regarding claim 22, Barker in view of Sampat fail to teach or suggest wherein the event distribution server comprises the plurality of event distribution server sinks. The Examiner admits that Barker does not teach a plurality event distribution server sinks, but cites figure 18, column 9, line 10 to column 10, line 46 and column 13, line 62 to column 14, line 57 of Sampat. However, none of the cited passages teaches or suggests an event distribution server comprising a plurality of event distribution server sinks. As discussed above regarding claim 1, Sampat teaches media service provider sinks that transfer media data to appropriate hardware device drivers and fails to teach a plurality of event distribution server sinks. For a more detailed discussion regarding Sampat's media

service provider sinks and Sampat's failure to teach or suggest event distribution server sinks, please see the remarks above regarding claims 1 and 14.

**Claims 24 and 37:**

Regarding claim 24, contrary to the Examiner's contention, Barker in view of Sampat fails to teach or suggest wherein the plurality of event distribution server sinks are operable to dispatch the events to one or more managers as a function of the subscription. The Examiner cites column 9, line 22 to column 10, line 49 and column 19, line 13 to column 20, line 59 of Barker. However, not only do neither of the cited passages mention anything regarding a plurality of event distribution server sinks operable to dispatch events to one or more managers as a function of subscriptions, **the Examiner has already stated, in the rejections of claims 1, 9, 14 and 22 that Barker fails to teach a plurality of event distribution server sinks** (see, e.g. Final Office Action dated February 14, 2005, page 4, line 1 and page 6, lines 1-2). Thus, the Examiner's assertion in the rejection of claim 11 is clearly improper. The Examiner cannot argue Barker both ways.

Furthermore, as discussed above, regarding the rejections of claims 1, 9, 14 and 22, Sampat fails to teach or suggest *event distribution* server sinks. Instead, Sampat teaches media service provider sinks that transfer media data to appropriate hardware device drivers. For a more detailed discussion regarding Sampat's media service provider sinks and Sampat's failure to teach or suggest event distribution server sinks, please refer to the above discussions regarding claims 1, 9, 14 and 22.

**Claims 25 and 38:**

Regarding claim 25, Barker in view of Sampat does not teach or suggest a plurality of event distribution server sinks that are distributed to provide load balancing of the events to the one or more managers. The Examiner cites col. 29, line 27 – col. 30, line 42; and col. 37, line 4 – col. 38, line 63, of Barker in regard to claim 25. However, these sections of Barker reveal absolutely nothing at all that corresponds to a plurality of

event distribution server sinks that are distributed to provide load balancing of the events to the one or more managers. In fact, the first cited section (Barker, col. 29, line 27 – col. 30, line 42) describes how Barker’s system provides overload control by limiting the amount of network traffic through only allowing a limited number of client sessions and running applications. This cited section also describes Barker’s preferred method of software version management. Neither of these topics discusses load balancing. The Examiner’s second cited passage (col. 37, line 4 – col. 38, line 63) describes the major functions of Barker’s event handler and goes on to provide overviews of Barker’s Network Element Status Table, Network Element Status API, Event Configuration File, Admin Log, and EMAPI, none of which mention (or have anything to do with) a plurality of event distribution server sinks that are distributing to provide load balancing of the events to the one or more managers.


For a more detailed discussion regarding Barker’s failure to teach event distribution server sinks that are distributed to provide load balancing of the events to the managers, please refer to Appellants’ arguments above regarding claim 12.

### VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-9, 11-22, 24-35 and 37-39 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of \$500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-48200/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,



Robert C. Kowert  
Reg. No. 39,255  
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
(512) 853-8850

Date: July 13, 2005



## **IX. CLAIMS APPENDIX**

The claims on appeal are as follows.

1. A network management system comprising:

an event gateway which is coupled to one or more managed objects and which is configured to deliver events generated by the managed objects to one or more managers; and

a platform-independent interface to the event gateway, wherein the event gateway is configurable to communicate with the managers through the platform-independent interface to deliver the events generated by the managed objects;

wherein the event gateway comprises a plurality of event distribution server sinks configured to receive events generated by the managed objects and distribute the events to the one or more managers such that one of the managers receives events from a plurality of different ones of the event distribution server sinks; and

wherein the gateway is configurable to provide the managers with subscriptions to the events as a function of event criteria specified by the managers, whereby events meeting the specified event criteria are delivered and events failing to meet the specified event criteria are filtered out.

2. The network management system of claim 1, wherein the event criteria comprise an object class for the managed objects generating the events.

3. The network management system of claim 1, wherein the event criteria comprise an object instance for one of the managed objects generating the events.

4. The network management system of claim 1, wherein the event criteria comprise an event type.

5. The network management system of claim 1, wherein the platform-independent interface to the event gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to managed objects across a plurality of platforms and across a plurality of programming languages.

6. The network management system of claim 5, wherein the interface definition language comprises Object Management Group Interface Definition Language (OMG IDL) interface.

7. The network management system of claim 1, wherein the managed objects comprise one or more objects corresponding to a telephone network.

8. The network management system of claim 1, wherein the managed objects comprise an object corresponding to a telecommunications device.

9. The network management system of claim 1, wherein the event gateway comprises:

an event distribution server, wherein the event distribution server is configurable to listen for the events generated by the one or more managed objects and deliver the events to the one or more managers, wherein the event distribution server comprises the plurality of event distribution server sinks.

11. The network management system of claim 9, wherein the event distribution server comprises:

an event distribution server source which listens for the events from the one or more managed objects; and

wherein the plurality of event distribution server sinks are operable to dispatch the events to the one or more managers as a function of the subscriptions.

12. The network management system of claim 11, wherein the event distribution server sinks are distributed to provide load balancing of the events to the one or more managers.

13. The network management system of claim 1, wherein the events are delivered through the platform-independent interface according to Internet Inter-Object Protocol (IIOP).

14. A network management method comprising:

registering a subscription of a manager application to one or more events generated by one or more managed objects by specifying event criteria to an event gateway, and wherein the event gateway is configurable to communicate with the manager application through a platform-independent interface;

generating a plurality of events including one or more events matching the specified event criteria;

determining whether the specified event criteria are met for each of the plurality of generated events; and

delivering each event for which the specified event criteria are met, wherein events for which the specified event criteria are met are delivered from a

plurality of different event distribution server sinks of the event gateway to the manager application.

15. The network management method of claim 14, wherein the event criteria comprise an object class for the managed objects generating the events.

16. The network management method of claim 14, wherein the event criteria comprise an object instance for one of the managed objects generating the events.

17. The network management method of claim 14, wherein the event criteria comprise an event type.

18. The network management method of claim 14, wherein the platform-independent interface to the event gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to managed objects across a plurality of platforms and across a plurality of programming languages.

19. The network management method of claim 18, wherein the interface definition language comprises Object Management Group Interface Definition Language (OMG IDL).

20. The network management method of claim 14, wherein the managed objects comprise one or more objects corresponding to a telephone network.

21. The network management method of claim 14, wherein the managed objects comprise an object corresponding to a telecommunications device.

22. The network management method of claim 14, wherein the event gateway comprises:

an event distribution server which is coupled to the event port registry server, wherein the event distribution server is configurable to listen for the events generated by the one or more managed objects and deliver the events to the one or more managers, wherein the event distribution server comprises the plurality of event distribution server sinks.

24. The network management method of claim 22, wherein the event distribution server comprises:

an event distribution server source which listens for the events from one or more managed objects; and

wherein the plurality of event distribution server sinks are operable to dispatch the events to managers as a function of the subscriptions.

25. The network management method of claim 24, wherein the event distribution server sinks are distributed to provide load balancing of the events to the one or more managers.

26. The network management method of claim 14, wherein the events are delivered through the platform-independent interface according to Internet Inter-Object Protocol (IIOP).

27. A carrier medium comprising program instructions for network management, wherein the program instructions are computer-executable to perform:

registering a subscription of a manager application to one or more events generated by one or more managed objects by specifying event criteria to an event gateway, and wherein the event gateway is configurable to communicate with the manager application through a platform-independent interface;

generating a plurality of events including one or more events matching the specified event criteria;

determining whether the specified event criteria are met for each of the plurality of generated events; and

delivering each event for which the specified event criteria are met, wherein events for which the specified event criteria are met are delivered from a plurality of different event distribution server sinks of the event gateway to the manager application.

28. The carrier medium of claim 27, wherein the event criteria comprise an object class for the managed objects generating the events.

29. The carrier medium of claim 27, wherein the event criteria comprise an object instance for one of the managed objects generating the events.

30. The carrier medium of claim 27, wherein the event criteria comprise an event type.

31. The carrier medium of claim 27, wherein the platform-independent interface to the event gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to managed objects across a plurality of platforms and across a plurality of programming languages.

32. The carrier medium of claim 31, wherein the interface definition language comprises Object Management Group Interface Definition Language (OMG IDL).

33. The carrier medium of claim 27, wherein the managed objects comprise one or more objects corresponding to a telephone network.

34. The carrier medium of claim 27, wherein the managed objects comprise an object corresponding to a telecommunications device.

35. The carrier medium of claim 27, wherein the event gateway comprises:

an event distribution server which is coupled to the event port registry server, wherein the event distribution server is configurable to listen for the events generated by the one or more managed objects and deliver the events to the one or more managers, wherein the event distribution server comprises the plurality of event distribution server sinks.

37. The carrier medium of claim 35, wherein the event distribution server comprises:

an event distribution server source which listens for the events from one or more managed objects; and

wherein the plurality of event distribution server sinks are operable to dispatch the events to managers as a function of the subscriptions.

38. The carrier medium of claim 37, wherein the event distribution server sinks are distributed to provide load balancing of the events to the one or more managers.

39. The carrier medium of claim 27, wherein the events are delivered through the platform-independent interface according to Internet Inter-Object Protocol (IIOP).

**X. EVIDENCE APPENDIX**

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.



**XI. RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.